Вестник Брестского государственного технического университета. 2013. №5
УДК 004.056

Волокита А.Н., Ву Дык Тхинь, Щербина А.В., Андресюк Б.Е., Бойкив Т.В., Паламарчук В.В.

МОДЕЛЬ МНОГОКАНАЛЬНОГО БЕЗОПАСНОГО ОБСЛУЖИВАНИЯ В PRIVATE CLOUD СИСТЕМЕ

Введение. Запуск задач с требованиями реального времени в Cloud системах накладывает определенный набор требований к уровню обслуживания задач, при этом использование стандартных алгоритмов планирования и обеспечения безопасности не позволяет удовлетворить данные требования.

В соответствии с последними докладами "Cloud Computing Synopisand Recommendations" Национального института стандартов и технологий США NIST, частное облако – это облачная инфраструктура, которая эксплуатируется исключительно организацией, может находиться под управлением организации или третьей стороны, и быть собственностью организации или третьей стороны [1].

При использовании собственного частного облака нет необходимости полагаться на внешние неподконтрольные сети и проще обеспечить внутреннюю безопасность. Но для создания и поддержки инфраструктуры требуются большие средства, чем при использовании сторонних сервисов. Кроме того, при внезапно нарастающей вычислительной нагрузке доступно меньшее количество ресурсов для расширения облака.

При использовании стороннего частного облака появляется гибкость в предоставлении дополнительных ресурсов в короткие сроки. Данная инфраструктура будет зависеть от внешних сетей, что потребует дополнительных усилий для обеспечения безопасности [2].

В частном облаке все ресурсы объединены в пулы, что позволяет достичь высокой эффективности и масштабируемости, за счет динамического изменения объема ресурсов, выделяемых под конкретную задачу. Распределяя ресурсы из общего пула между несколькими задачами и пользователями, можно повысить эффективную загрузку имеющихся в наличии ресурсов, фактически это позволяет быстро масштабировать сервисы в соответствии с требованиями клиентов.

При запросе, настройке и управлении поставщики услуг и потребители используют интерактивный портал или систему, предназначенную для автоматического предоставления ресурсов [3].

Модель многоканального обслуживания в Private Cloud. Для удовлетворения требований реального времени в Cloud системах предложена модель системы реального времени (СРВ), в которой применяется программный модуль на основе SSA (secret sharing algorithm) для разделения и сборки данных при многоканальной (N каналов) передаче.

Клиенты разделяются на группы по уровню безопасности и приоритету пользователя. Для клиентов, которым требуется уровень безопасности выше минимального, предлагается использовать несколько каналов связи с облаком, для передачи частей задачи по разным линиям.

Клиентам с более высоким уровнем приоритета гарантируется большая вероятность безотказной работы.

Для клиентов с уровнем безопасности выше минимального задача разделяется по алгоритму SSA на N частей и по разным каналам доставляется во входную очередь, где происходит сборка. Если пользователь выбирает минимальный уровень безопасности, то задача доставляется по одному каналу связи.

Во входной очереди накапливаются задачи на выполнение. Каждая задача имеет следующие параметры: приоритет, уровень безопасности, требования к оборудованию (объем памяти, частота процессора), время выполнения на требуемом оборудовании, время, за которое должен быть получен результат. Через заданный в системе администратором промежуток времени или после накопления определенного количества задач во входной очереди выполняется планирование распределения задач по ресурсам.

Каждый ресурс системы имеет следующий набор параметров: объем памяти, количество ядер, частота процессора и уровень безопасности, который определяется степенью защиты виртуальных машин.

После планирования задачи доставляются на каналы обслуживания, где происходит собственно выполнение заданий, затем обработанные результаты доставляются в выходную очередь. Результаты задач с минимальным уровнем безопасности из выходной очереди непосредственно доставляются клиентам, а остальные результаты разбиваются на части по алгоритму SSA и доставляются клиентам по нескольким каналам связи.

Модифицированный алгоритм планирования A* для Cloud системы реального времени. A*(A звездочка) – алгоритм поиска по первому наилучшему совпадению на графе, для поиска маршрута с наименьшей стоимостью от одной вершины (начальной) к другой (целевой, конечной) [4].

Порядок обхода вершин определяется эвристической функцией «расстояние + стоимость» (обычно обозначаемой как $f(x)$). Эта функция – сумма двух других: функции $g(x)$ стоимости достижения рассматриваемой вершины (x) из начальной и эвристической оценки $h(x)$ расстояния от рассматриваемой к конечной вершине. Поиск продолжается до тех пор, пока не будет выбран узел с полным назначением.

Для тестирования модели Private Cloud системы выбран данный алгоритм, так как обеспечивает высокое качество планирования и позволяет гибко учитывать требования задач к реальному времени.

Для работы в условиях реального времени алгоритм модифицирован.

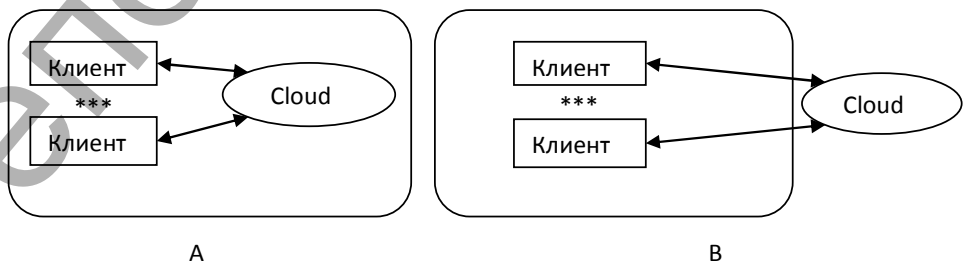

Рис. 1. A – Собственное частное облако, B – стороннее частное облако

Волокита Артем Николаевич, *к.т.н., доцент кафедры вычислительной техники Национального технического университета Украины "Киевский политехнический институт".*
Ву Дык Тхинь, *аспирант кафедры вычислительной техники Национального технического университета Украины "Киевский политехнический институт".*
Щербина А.В., Андресюк Б.Е., Бойкив Т.В., Паламарчук В.В., *студенты Национального технического университета Украины "Киевский политехнический институт".*
Украина, г. Киев, пр. Победы, 37, e-mail: artem.volokita@kpi.ua.

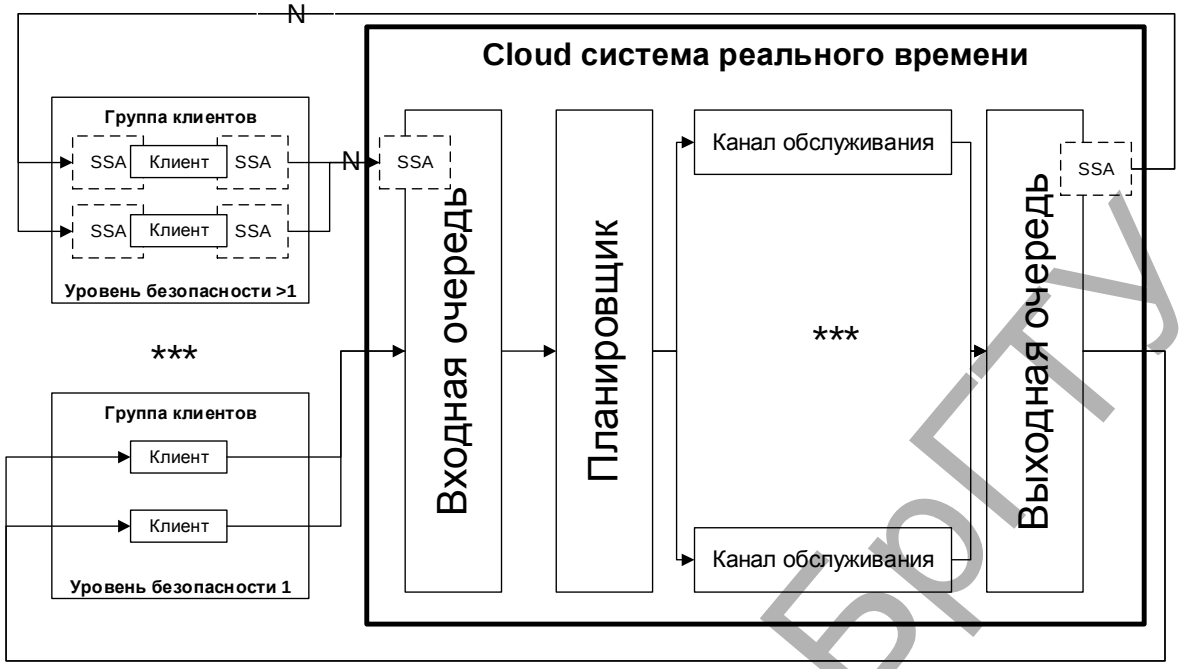

Рис. 2. Модель Private Cloud системы

Каждый ресурс имеет коэффициент загрузки k_{load} в диапазоне [0, 1] – процент занятого процессорного времени. При загрузке задачи на ресурс, рассчитывается, какой процент k'_{load} процессорного времени требует задача. Данный коэффициент вводится для того, чтобы избежать возможной перегрузки узла, поскольку при перегрузке не гарантируется заданное время отклика системы.

Если $k_{load}+k'_{load}>1$, то ресурс будет перегружен и не сможет соответствовать требованиям реального времени, поэтому выполнение данной задачи на этом ресурсе невозможно. При этом значение функции стоимости для данного узла устанавливается равным бесконечности.

Иначе коэффициент загрузки ресурса, после погружения задачи на этот ресурс, будет вычисляться следующим образом: $k_{load}:=k_{load}+k'_{load}$.

Также учитываются уровни безопасности задачи и соответствующего ресурса. Если уровень безопасности ресурса выше или равен уровню безопасности задачи, то выполнение задачи на ресурсе возможно, иначе ресурс для задачи не подходит и значение функции стоимости устанавливается в бесконечность. Обеспечивается требуемый уровень безопасности задачи во время выполнения.

Таким образом, модификация алгоритма планирования позволяет избежать перегрузки системы и обеспечивает требуемый уровень безопасности при выполнении задач.

Разделение данных на основе алгоритма Шамира для безопасной многоканальной передачи. Для безопасной передачи данных по открытым каналам связи будем использовать разделение данных на части. Для восстановления исходного сообщения, разделенного на n частей, необходимо собрать k частей, причем $k \le n$. Таким образом, безопасность передачи существенно повышается.

В алгоритме разделения секрета Блекли [5] задается размерность пространства, равная n, и по каждому из n каналов передается одна гиперплоскость, которая проходит через секретную точку M. Тогда любые k из n гиперплоскостей будут однозначно пересекаться в секретной точке. В данном алгоритме используются простые числа: выбирается простое число p, большее M. Затем выбираются числа меньше p: $d_1, d_2, \ldots, d_n$, для которых значения d_i упорядочены по возрастанию; каждое d_i взаимно-простое с другими d_i.

Для пороговой схемы $(n; k)$ требуется выполнение неравенства:

$$d_1 * d_2 * \ldots * d_k > p * d_n - k + 2 * d_n - k + 3 * \ldots * d_n.$$

Чтобы распределить части, сначала выбирается случайное число r и вычисляется $M' = M + rp$. Тенями k_i являются $k_i = M' \bmod d_i$, где тени – это проекции точки на гиперплоскость.

Объединив любые k частей, можно восстановить M, используя теорему об остатках, однако это невозможно сделать, используя лишь $k-1$ частей.

В алгоритме Karnin-Greene-Hellman используется матричное умножение. Выбирается $n+1$ k-мерных векторов $V_0, V_1, \ldots, V_n$ так, что ранг любой матрицы размером $k*k$, образованный из этих векторов равен k. Вектор U это вектор размерности $k+1$. Секрет M – это матричное произведение $U*V_0$. Частями секрета являются произведения $U*V_i$, где i меняется от 1 до k. Любые k частей можно использовать для решения системы линейных уравнений размерности $k*k$, в которой неизвестными являются коэффициенты U. Таким образом, секрет $U*V_0$ можно вычислить, только зная вектор U. Если известно только $k-1$ часть, то решить систему уравнений и раскрыть секрет невозможно.

Идея алгоритма Shamir [6] заключается в том, что двух точек достаточно для задания прямой, трех точек – для задания параболы, четырёх точек – для кубической параболы, и так далее. Чтобы задать многочлен степени k требуется $k+1$ точек.

Для разделения секрета, чтобы восстановление было возможно только при наличии не меньше чем k частей, данные трансформируются в многочлен, который можно восстановить по точкам.

Алгоритм Шамира более эффективен, чем приведенные выше алгоритмы, за счет того, что часть сообщения для каждого из каналов такого же размера как и секрет, а в алгоритме Блекли каждая часть в k раз больше секрета. Поэтому в роботе применяется алгоритм разделения секрета Шамира, рассмотренный ниже более детально.

Выберем некоторое простое число $p>M$. Это число открыто сообщается всем участникам и задаёт конечное поле размера p. Над этим полем строится многочлен степени $k-1$, то есть случайно выбираются все коэффициенты многочлена, кроме M:

$$F(x) = \left(a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \ldots + a_1 x + M\right) \bmod p,$$

где M – это разделяемый секрет, а коэффициенты $a_{k-1}, a_{k-2}, \ldots, a_1$ – некоторые случайные числа, которые уничтожаются после того, как процедура разделения секрета будет завершена.

Затем вычисляются координаты различных точек:

$$k_i = F(i) = \left(a_{k-1} i^{k-1} + a_{k-2} i^{k-2} + ... + a_1 i + M\right) mod\ p\ .$$

При этом номера секретов должны быть различны по модулю p.

После этого части секрета вместе с их номерами, числом и степенью многочлена передаются по разным каналам. Теперь любые участники, зная координаты k различных точек многочлена, смогут восстановить многочлен и все его коэффициенты, включая последний из них – разделённый секрет.

Прямолинейное восстановление коэффициентов многочлена через решение системы уравнений подменяется вычислением интерполяционного многочлена Лагранжа. Формула многочлена выглядит следующим образом:

$$F(x) = \sum_i l_i(x) y_i \ mod\ p;$$

$$l_i(x) = \prod_{i \neq j} \frac{x - x_j}{x_i - x_j} mod\ p,$$

где x_i, x_j – координаты точек многочлена, и все операции выполняются в конечном поле размера p.

Для обеспечения безопасности доставки задач в Cloud систему и доставки результатов клиенту использован алгоритм разделения секрета Шамира. Для этого на клиенте устанавливаются программные средства для разделения и отправки задач по n каналам связи во входную очередь Cloud системы, где выполняется восстановление из фрагментов.

Моделирование private cloud системы реального времени. Основной целью проводимого вычислительного эксперимента было исследование влияния многоканальности на работу облачной системы реального времени. Ограниченный набор воздействий и исследуемых характеристик создает предпосылки для упрощения используемой модели и соответственно уменьшения ресурсов необходимых для моделирования. При этом точность определяемых параметров ухудшается незначительно.

Разработано программное обеспечение для моделирования многоканального безопасного обслуживания в private cloud системе реального времени. Данное ПО функционирует в облаке, что позволяет гибко настраивать вычислительные мощности для моделирования.

Моделирование входного потока заявок ограничено100 потоками Эрланга, каждый из которых моделирует поведения одного пользователя, что связано с ограничением вычислительных ресурсов.

На основе распределения Эрланга генерировалось поступление задач во входную очередь и параметры этих задач. Модель PrivateCloud системы состоит из 20 ресурсов, каждый из которых содержит от 1 до 4 процессоров.

Измерялись такие характеристики системы:

- зависимость времени ожидания завершения выполнения задачи от длины входной очереди;
- зависимость загрузки системы от длины входной очереди;
- зависимость количества отброшенных задач от длины входной очереди;
- зависимость вероятности отказа задачи от размера входной очереди;
- зависимость времени ожидания задачи от уровня приоритета задачи.

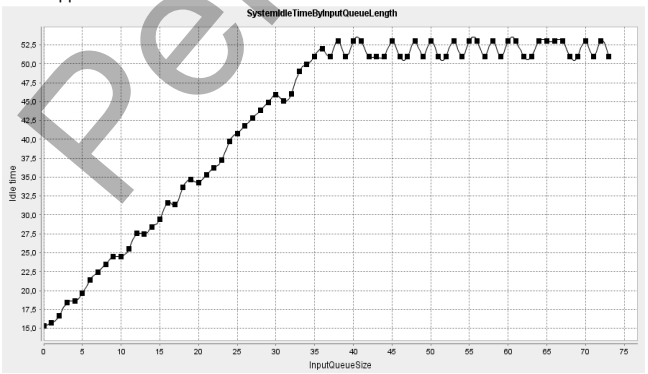

Рис. 3. Зависимость времени ожидания завершения задачи от длины входной очереди

Как видно из рис. 3, время ожидания задачи растет вместе с длиной входной очереди, что объясняется увеличением времени простоя задачи в очереди и большей степенью загрузки узлов. При достижении полной загрузки, отбрасываются новые входящие задачи, поскольку при приеме большего количества задач система не сможет удовлетворять требованиям реального времени. При этом минимальное время ожидания завершения равно времени выполнения задачи.

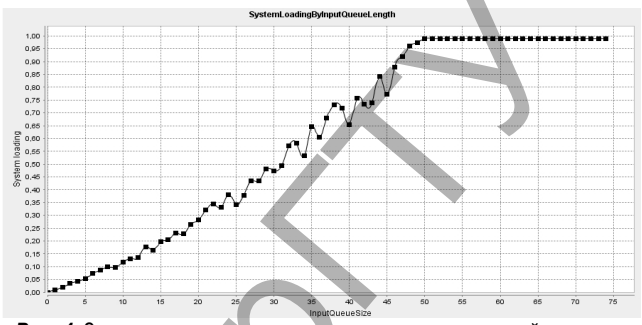

Рис. 4. Зависимость загрузки системы от длины входной очереди

Из рис. 4 видно, что при увеличении количества задач во входной очереди, загрузка системы растет, и при полной загрузке задачи начинают отбрасываться. За счет отказа в облуживании при достижении пиковой нагрузки обеспечиваются требования реального времени.

Нелинейность роста загрузки системы объясняется тем, что скорость обработки меньше скорости поступления задачи из-за продолжительного времени их выполнения.

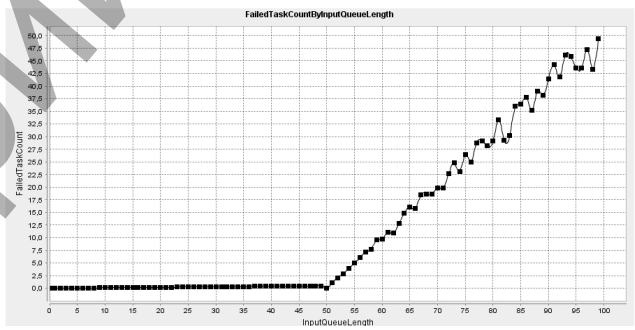

Рис. 5. Зависимость количества отброшенных задач от длины входной очереди

Как видно из рис. 5, при количестве задач во входной очереди $N_{вход\ пороговое}$~*50*, система обрабатывает все задания. После превышения $N_{вход\ пороговое}$ количество отбрасываемых задач возрастает линейно и равно $N_{вход}$-$N_{вход\ пороговое}$, где $N_{вход}$ – количество задач во входной очереди. Это объясняется тем, что при количестве задач во входной очереди, достигающем $N_{вход\ пороговое}$, наступает 100% загрузка системы. Количество задач для полной загрузки системы можно определить из рис. 4.

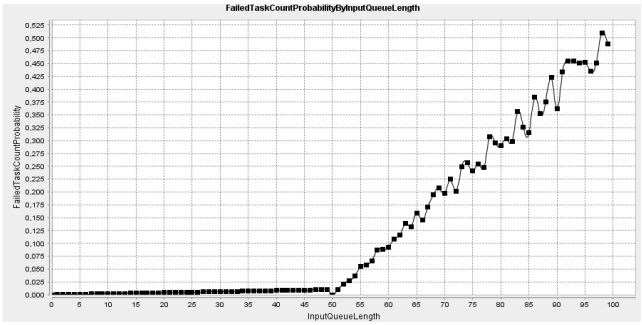

Рис. 6. Зависимость вероятности отброса задачи от размера входной очереди

Вероятность отказа p рассчитывается по формуле $p = \frac{K}{M+K}$, где K – количество отброшенных задач, M – количество выполненных задач. Как показано выше, при количестве задач во входной очереди меньше $N_{входпороговое}$, вероятность отброса задачи равна 0, затем график вероятности растет линейно с ростом количества задач во входной очереди.

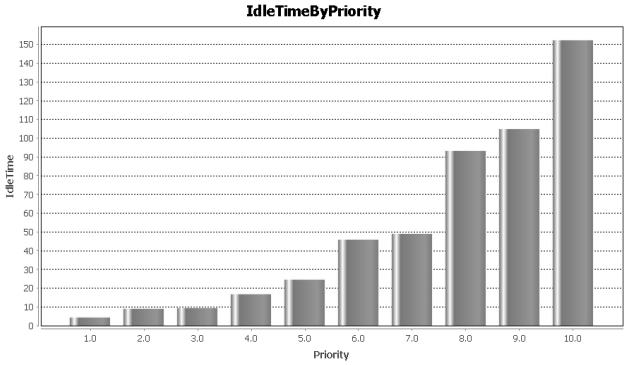

Рис. 7. Зависимость времени ожидания задачи от уровня приоритета задачи

Как видно из рис. 7, задачи с более высоким уровнем приоритета находятся в режиме ожидания меньше, чем задачи с низким приоритетом. Чем выше загрузка системы, тем выше время ожидания для низкоприоритетных задач.

Созданная модель PrivateCloud системы реального времени доступна по адресу https://github.com/sansherbina/RealTimeSchedulingSystemModel. Моделирование проводилось в облачной инфраструктуре, что позволило справляться с высокими требованиями к вычислительным ресурсам и динамической нагрузкой.

Результаты моделирования показали, что Cloud система в нормальном режиме (с уровнем загрузки меньше 100%) удовлетворяет требованиям реального времени. Как видно из рис. 6, вероятность отброса задачи приблизительно равна 0 до достижением системой пика загрузки. Также разработанная система приоритетов позволяет эффективно дифференцировать пользователей по уровню обслуживания. Как видно из рисунка 7, задачи с уровнем 1 (наиболее приоритетный уровень) имеет наиболее низкое время ожидания.

Заключение. В данной статье модифицирован для работы в Cloud системе реального времени алгоритм планирования выполнения задач A*. Программный модуль выполняет функции распределения задач и балансировки нагрузки в частном облаке. При этом дополнительно возможно выбирать уровень безопасности, что позволяет защитить данные в момент выполнения задачи.

Для обеспечения необходимого уровня безопасности при передаче задач от пользователя в систему и результатов обратно использован алгоритм разделения секрета SSA. Задача и данные разделяются на части по алгоритму Shamir и передаются в Cloud систему по отдельным каналам, что позволяет повысить уровень безопасности при передаче.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. National Institute of Standards and Technology. [Электронный ресурс]. – Режим доступа: http://www.nist.gov/index.html. – Дата доступа: 10.04.2013.
2. Smart Computing in Real Time. [Электронный ресурс]. – Режим доступа: http://www.real-timecloud.com. – Дата доступа: 20.03.2013.
3. Cloud Computing Journal. [Электронный ресурс]. – Режим доступа: http://cloudcomputing.sys-con.com. – Дата доступа: 30.03.2013.
4. Muhammad Kafil and Ishfaq Ahmad // Optimal Task Assignment in Heterogeneous Distributed Computing Systems. The Hong Kong University of Science and Technology. 2011.
5. Bruce Schneier. Applied Cryptography. Protocols, Algorithms and Source Code in C. – М.: Триумф, 2002. – С. 589–816.
6. Shamir, Adi. How to share a secret // Communications of the ACM, 1979.

Материал поступил в редакцию 19.11.13

VOLOKITA A.N., VU DUK THIN, SHCHERBYNA A.V., ANDRESYUK B.E., BOYKIV T.V., PALAMARCHUK V.V. Model of multichannel secure service in Private Cloud

In this paper proposed an improvement of the scheduling algorithm A* for using in Private Cloud real-time systems. Algorithm in conjunction with the algorithm of Shamir secret sharing allows static scheduling with the high level of security and the required response time for tasks. Ensured safety delivering tasks to the Cloud system, and the results back to the user.

УДК 535.337

Русаков К.И., Ракович Ю.П., Гладыщук А.А., Мельников Д.Г., Саватеева Д.И., Русакова З.В., Чугунов С.В.

ОПТИЧЕСКИЕ ПРОЦЕССЫ В МИКРОРЕЗОНАТОРАХ С J-АГРЕГАТАМИ

Введение. Одним из основных направлений в оптике микрорезонаторов является создание эффективной связи электронных переходов в органических и неорганических наноструктурах с фотонными модами резонаторов. Диэлектрические микросферы являются трехмерными микрорезонаторами с большой добротностью и малым объемом мод, что приводит к возникновению в них сильной оптической обратной связи с резонатором [1]. Оптические резонансы этих резонаторов, называемые модами шепчущей галереи, возникают вследствие полного внутреннего отражения света от внутренней поверхности сферы. Резонаторы мод шепчущей галереи (МШГ) представляют интерес как для изучения взаимодействия света с веществом [2], так и для разного рода практических применений, как низкопороговые лазеры [3], устройства динамического фильтрования в волоконной оптике [4] и оптические сенсоры [5]. Использование сферических микрорезонаторов может быть расширено за счет различных нелинейных оптических эффектов при малых интенсивностях накачки. Ранее были исследованы МШГ в стеклянных и полимерных микросферах, интегрированных с неорганическими люминесцирующими материалами [2].

Русаков Константин Иванович, *профессор кафедры физики Брестского государственного технического университета.*
Гладыщук Анатолий Антонович, *заведующий кафедрой физики Брестского государственного технического университета.*
Русакова Зоя Витальевна, *старший преподаватель кафедры физики Брестского государственного технического университета.*
Чугунов Сергей Владимирович, *старший преподаватель кафедры физики Брестского государственного технического университета.*
Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.
Ракович Юрий Петрович, *профессор-исследователь Центра физики материалов, Сан-Себастьян, Испания.*
Мельников Дмитрий Георгиевич, *научный сотрудник Центра физики материалов, Сан-Себастьян, Испания.*
Саватеева Диана Игоревна, *научный сотрудник Центра физики материалов, Сан-Себастьян, Испания.*